INDIAN INSTITUTE OF SPACE SCIENCE AND TECHNOLOGY
THIRUVANANTHAPURAM

Assignment #5

Due on 19-11-2014

SUHAS S
(SC14081)

Contents

1.K-means & K-medioid

1.1 K-means

a.using Eucledian distance

K value	DB index
2	3.838935
3	1.191889
4	1.235967
5	1.285125
6	1.226606

b.using Manhattan distance

K value	DB index
2	3.615385
3	1.197559
4	2.108696
5	1.753846
6	1.294872

Plot of J(c,μ)

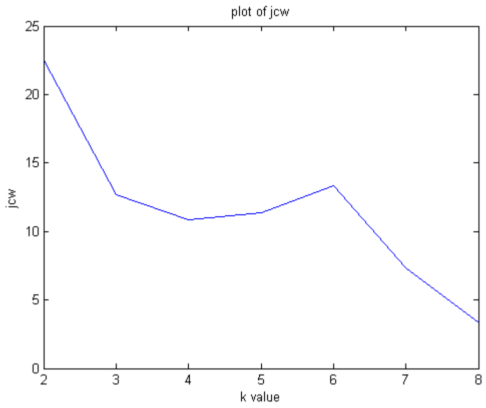

Figure 1: J(c,μ) v/s K value

The clusters obtained are(same for both Eucledian & Manhattan Metric)

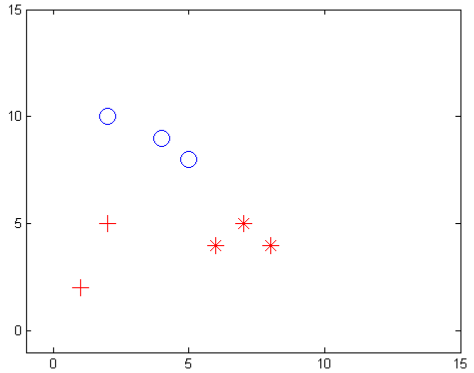

Figure 2: Clusters with k-means

1.2 K-medioid

a.using Eucledian distance

K value	DB index
2	3.077485
3	0.957000
4	1.447470
5	1.022754
6	1.147529

b.using Manhattan distance

K value	DB index
2	3.000000
3	0.975000
4	1.566667
5	1.034286
6	1.177778

Plot of J(c, μ)

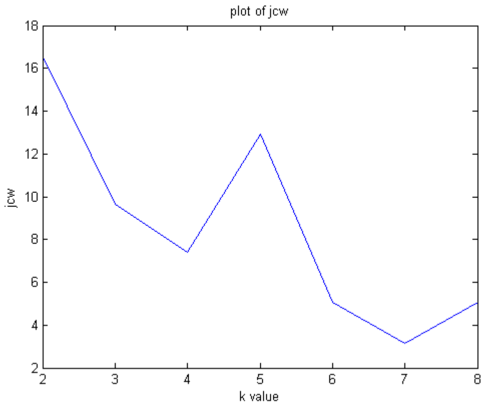

Figure 3: J(c,μ) v/s K value

The clusters obtained are(same for both Eucledian & Manhattan Metric)

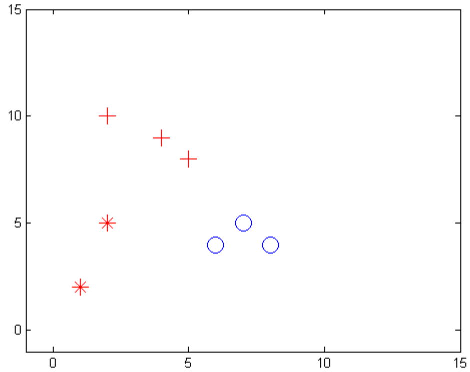

Figure 4: Clusters with k-means

2.Agglomerative & Divisive Clustering

2.1 Agglomerative Clustering

The merging sequence is

- (A_3),(A_5)
- (A_4),(A_8)
- $(A_3$,$A_5)$,(A_6)
- (A_1),$(A_4$,$A_8)$
- (A_2),(A_7)
- $(A_3$,A_5,$A_6)$,(A_1, A_4, A_8)
- $(A_3, A_5, A_6, A_1, A_4, A_8)$,$(A_2, A_7)$

the corresponging dendogram is

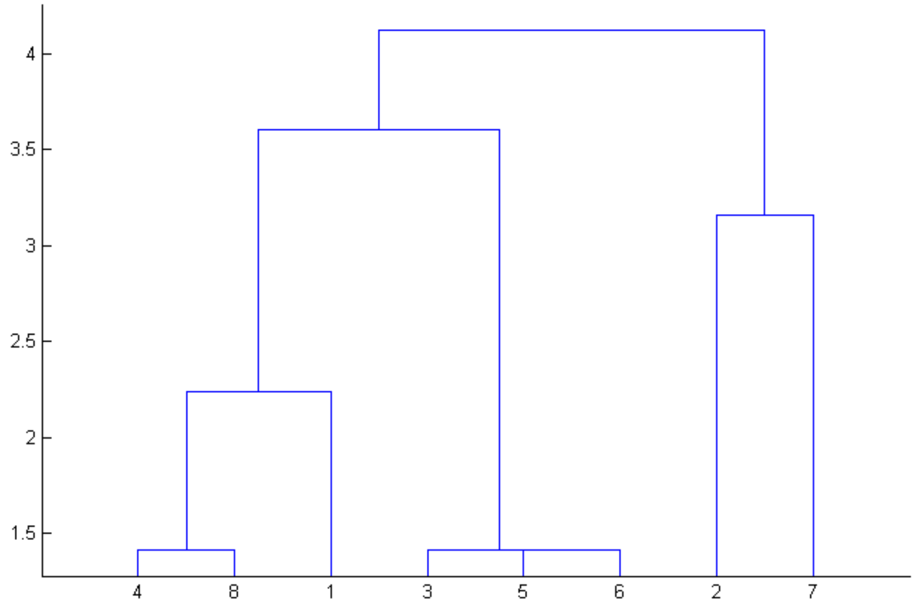

Figure 5: Agglomerative clustering

2.2 Divisive Clustering

Here the splitting sequence is

- L1 (A_1, A_4, A_8),$(A_2, A_3, A_5, A_6, A_7)$
- L2 (A_1),(A_4, A_8),(A_2, A_7),(A_3, A_5, A_6)
- L3 (A_4),(A_8),(A_2),(A_7),(A_3, A_5),(A_6)
- L4 (A_3),(A_5)

The splitting at a level is done by performing k-means clustering at that level with k=2.

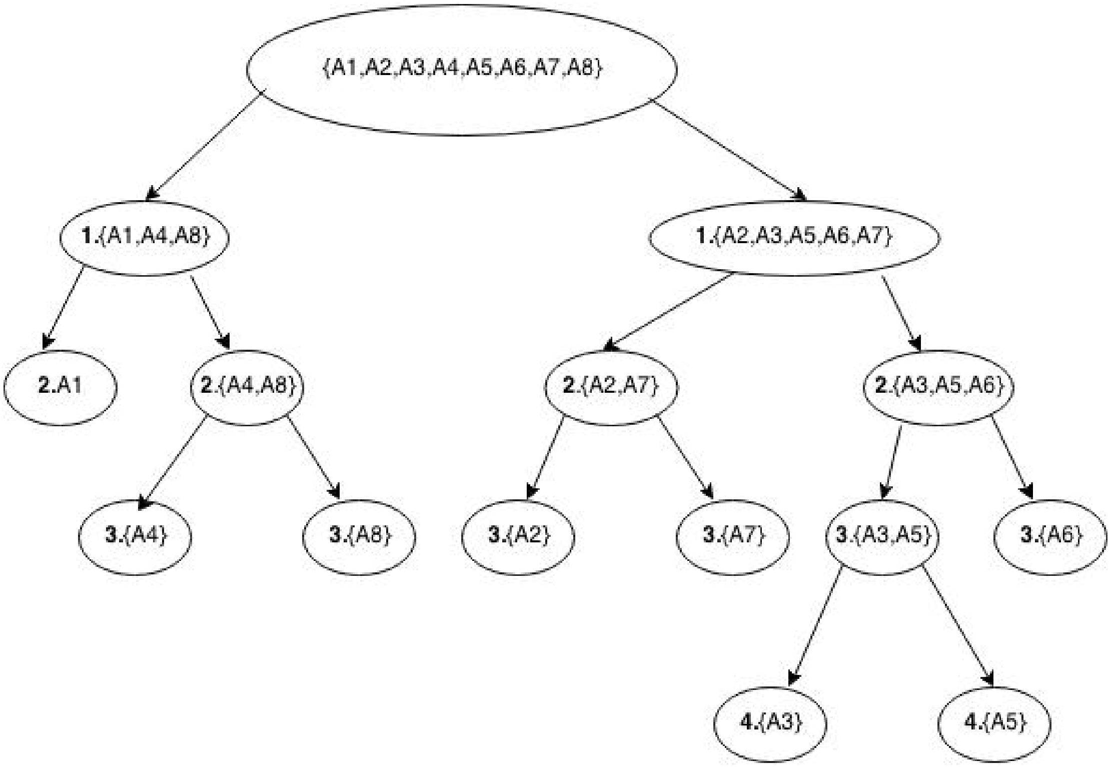

Figure 6: Divisive clustering

3.Self Organizing Maps

Best classification obtained(from multiple runs of the algorithm)

fraction of class1 flowers:0.344
fraction of class2 flowers:0.344
fraction of class3 flowers:0.313

Since the fraction of each class of flowers is 0.333 in the give data set this can be thought of as a good classification.

SOM Neurons when initialized

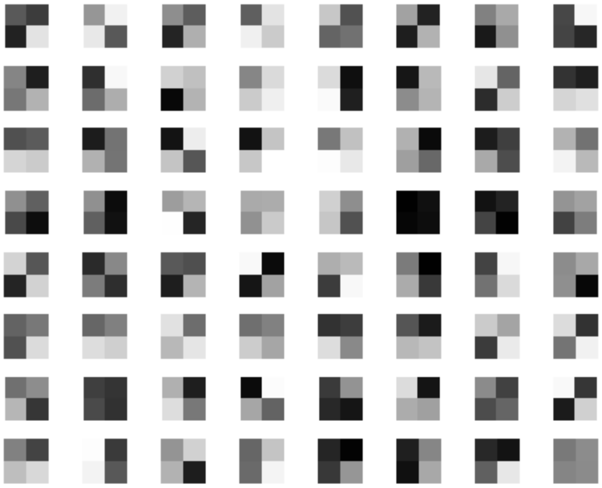

Figure 7: Initial state

SOM Neurons after training is over

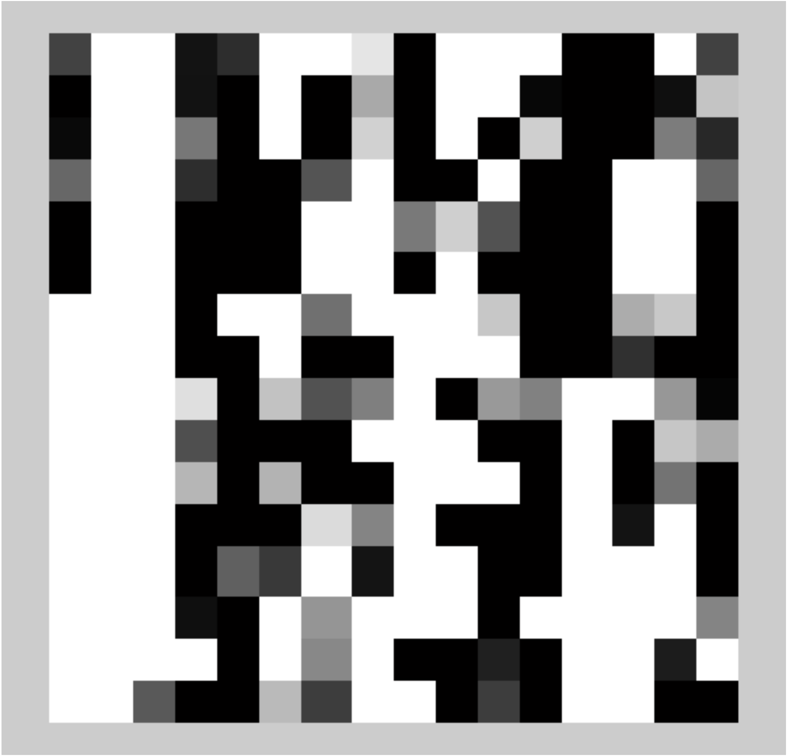

Figure 8: Convergence status

As we can see, the white regions belongs to clusters and the dark border is the one which separates the clusters from one another.

Hyper Parameter values:
α=0.05
No. of clusters = 3

4.Visualization Techniques in SOM

4.1 U-matrix

The U-matrix stands for unified distance and contains in each cell the euclidean distance (in the input space) between neighboring cells. Small values in this matrix mean that SOM nodes are close together in the input space, whereas larger values mean that SOM nodes are far apart, even if they are close in the output space. As such, the U-matrix can be seen as summary of the probability density function of the input matrix in a 2D space. Usually, those distance values are discretized, color-coded based on intensity and displayed as a kind of heatmap.
U-matrix (unified distance matrix) representation of the Self-Organizing Map visualizes the distances between the neurons. The distance between the adjacent neuons is calculated and presented with different colorings between the adjacent nodes. A dark coloring between the neurons corresponds to a large distance . A light coloring between the neurons signifies that the vectors are close to each other in the input space. Light areas can be thought as clusters and dark areas as cluster separators. This can be a helpful presentation when one tries to find clusters in the input data without having any a priori information about the clusters.
The U-Matrix value of a particular node is the average distance between the node's weight vector and that of its closest neighbors. In a square grid, for instance, we might consider the closest 4 or 8 nodes (the Von Neumann and Moore neighborhoods, respectively), or six nodes in a hexagonal grid[4].

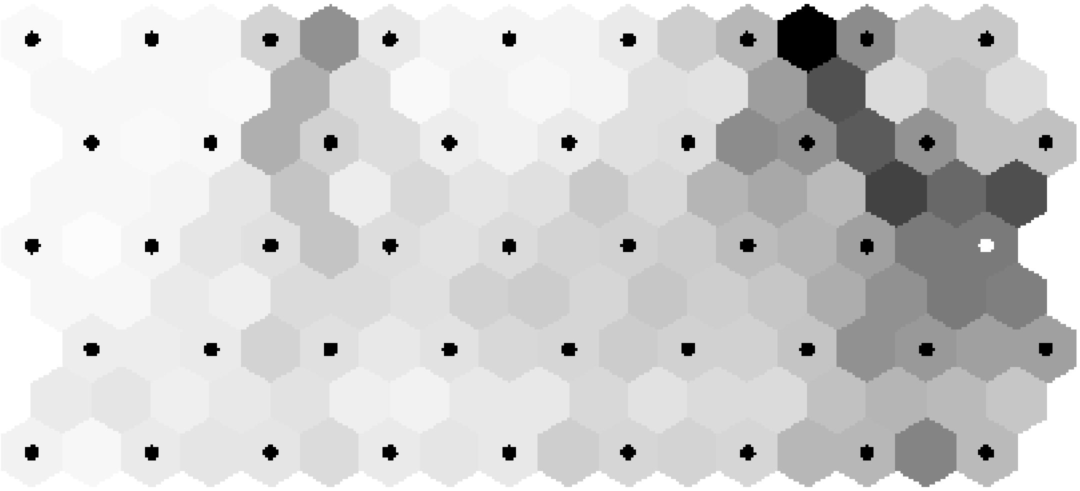

Figure 9: U-matrix

In the Figure above we can see the neurons of the network marked as black dots. The representation reveals that these are a separate cluster in the upper right corner of this representation. The clusters are separated by a dark gap.

4.2 Other Methods

There are many other methods proposed by researchers which uses visualization techniques that take the distribution of the data set in input space and its density into account. Most commonly, this is visualized as hit histograms, which display the number of data points projected to each map node. A more advanced method is the P-Matrix that visualizes how densely populated each unit is by counting the number of data points within the sphere of a certain radius around the model vector in question. Another recently proposed technique that aims at depicting both density and cluster structures is the Smoothed Data Histogram, which relies on a parameter that determines how blurred the visualization will be. There are also techniques that depict the contribution of the individual variable dimensions to the clustering structure, like LabelSOM. Other techniques providing insight into the distribution of the data manifold are projection methods like PCA and Sammon‘s Mapping[3].

5.Cluster Evaluation Techniques

Cluster evaluation techniques are mainly classified into two.

- Internal evaluation
- Exrernal evaluation

5.1 Internal Evaluation

When a clustering result is evaluated based on the data that was clustered itself, this is called internal evaluation. These methods usually assign the best score to the algorithm that produces clusters with high similarity within a cluster and low similarity between clusters. One drawback of using internal criteria in cluster evaluation is that high scores on an internal measure do not necessarily result in effective information retrieval applications.Additionally, this evaluation is biased towards algorithms that use the same cluster model. For example k-Means clustering naturally optimizes object distances, and a distance-based internal criterion will likely overrate the resulting clustering.Therefore, the internal evaluation measures are best suited to get some insight into situations where one algorithm performs better than another, but this shall not imply that one algorithm produces more valid results than another[2].

a.Davies-Bouldin index
The Davies-Bouldin index can be calculated by the following formula:

$$DB = \frac{1}{n}\sum_{i=1}^{n}\max_{i\neq j}\left(\frac{\sigma_i+\sigma_j}{d(c_i,c_j)}\right)$$

where n is the number of clusters, c_x is the centroid of cluster x, σ_x is the average distance of all elements in cluster x to centroid c_x, and $d(c_i,c_j)$ is the distance between centroids c_i and c_j. Since algorithms that produce clusters with low intra-cluster distances (high intra-cluster similarity) and high inter-cluster distances (low inter-cluster similarity) will have a low Davies-Bouldin index, the clustering algorithm that produces a collection of clusters with the smallest Davies-Bouldin index is considered the best algorithm based on this criterion.

b.Dunn Index
The Dunn index aims to identify dense and well-separated clusters. It is defined as the ratio between the minimal inter-cluster distance to maximal intra-cluster distance. For each cluster partition, the Dunn index can be calculated by the following formula:

$$D = \min_{1\leq i\leq n}\left\{\min_{1\leq j\leq n, i\neq j}\left\{\frac{d(i,j)}{\max_{1\leq k\leq n} d'(k)}\right\}\right\}$$

where *d(i,j)* represents the distance between clusters *i* and *j*, and $d^{'}(k)$ measures the intra-cluster distance of cluster k. The inter-cluster distance *d(i,j)* between two clusters may be any number of distance measures, such as the distance between the centroids of the clusters. Similarly, the intra-cluster distance $d^{'}(k)$ may be measured in a variety ways, such as the maximal distance between any pair of elements in cluster k. Since internal criterion seek clusters with high intra-cluster similarity and low inter-cluster similarity, algorithms that produce clusters with high Dunn index are more desirable.

c.Silhouette coefficient

The silhouette coefficient contrasts the average distance to elements in the same cluster with the average distance to elements in other clusters. Objects with a high silhouette value are considered well clustered, objects with a low value may be outliers. This index works well with k-means clustering, and is also used to determine the optimal number of clusters.

5.2 External Evaluation

In external evaluation, clustering results are evaluated based on data that was not used for clustering, such as known class labels and external benchmarks. Such benchmarks consist of a set of pre-classified items, and these sets are often created by human (experts). Thus, the benchmark sets can be thought of as a gold standard for evaluation. These types of evaluation methods measure how close the clustering is to the predetermined benchmark classes. However, it has recently been discussed whether this is adequate for real data, or only on synthetic data sets with a factual ground truth[2].

a.Rand Measure

The Rand index computes how similar the clusters (returned by the clustering algorithm) are to the benchmark classifications. One can also view the Rand index as a measure of the percentage of correct decisions made by the algorithm. It can be computed using the following formula:

$$RI = \frac{TP + TN}{TP + FP + FN + TN}$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives. One issue with the Rand index is that false positives and false negatives are equally weighted. This may be an undesirable characteristic for some clustering applications.

b.F-measure

The F-measure can be used to balance the contribution of false negatives by weighting recall through a parameter $\beta \geq 0$. Let precision and recall be defined as follows:

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

where P is the precision rate and R is the recall rate. We can calculate the F-measure by using the following formula:

$$F_\beta = \frac{(\beta^2 + 1) \cdot P \cdot R}{\beta^2 \cdot P + R}$$

Notice that when β=0, F_0=P. In other words, recall has no impact on the F-measure when β=0, and increasing β allocates an increasing amount of weight to recall in the final F-measure.

c. Jaccard Index
The Jaccard index is used to quantify the similarity between two datasets. The Jaccard index takes on a value between 0 and 1. An index of 1 means that the two dataset are identical, and an index of 0 indicates that the datasets have no common elements. The Jaccard index is defined by the following formula:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{TP}{TP+FP+FN}$$

This is simply the number of unique elements common to both sets divided by the total number of unique elements in both sets.

d. Confusion Matrix
A confusion matrix can be used to quickly visualize the results of a classification (or clustering) algorithm. It shows how different a cluster is from the gold standard cluster.

e. Purity
Purity is a simple and transparent evaluation measure.To compute purity , each cluster is assigned to the class which is most frequent in the cluster, and then the accuracy of this assignment is measured by counting the number of correctly assigned documents and dividing by N. Formally:

$$(\Omega, \mathbb{C}) = \frac{1}{N} \sum_k \max_j |\omega_k \cap c_j|$$

where $\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$ is the set of clusters and $\mathbb{C} = \{c_1, c_2, \ldots, c_J\}$ is the set of classes.
Bad clusterings have purity values close to 0, a perfect clustering has a purity of 1.High purity is easy to achieve when the number of clusters is large - in particular, purity is 1 if each document gets its own cluster. Thus, we cannot use purity to trade off the quality of the clustering against the number of clusters[1].

6.Mahalanobis Distance

Mahalanobis distance measures the distance of a point x from a data distribution. The data distribution is characterized by a mean and the covariance matrix, thus is hypothesized as a multivariate gaussian.The covariance matrix gives the shape of how data is distributed in the feature space.
The Mahalanobis distance has the following properties:

- It accounts for the fact that the variances in each direction are different.
- It accounts for the covariance between variables.
- It reduces to the familiar Euclidean distance for uncorrelated variables with unit variance.

For univariate normal data, the univariate z-score standardizes the distribution (so that it has mean 0 and unit variance) and gives a dimensionless quantity that specifies the distance from an observation to the mean in terms of the scale of the data. For multivariate normal data with mean μ and covariance matrix Σ, you can decorrelate the variables and standardize the distribution by applying the Cholesky transformation z = L^{-1}(x - μ), where L is the Cholesky factor of Σ, Σ=LL^T.

After transforming the data, we can compute the standard Euclidian distance from the point z to the origin. In order to get rid of square roots, we will compute the square of the Euclidean distance, which is dist2(z,0) = $z^T z$. This measures how far from the origin a point is, and it is the multivariate generalization of a z-score. We can rewrite $z^T z$ in terms of the original correlated variables. The squared distance $Mahal^2$(x,μ) is

$$= z^T z$$

$$= (L^{-1}(x-\mu))^T (L^{-1}(x-\mu))$$
$$= (x-\mu)^T (LL^T)^{-1}(x-\mu)$$
$$= (x-\mu)^T \Sigma^{-1}(x-\mu)$$

The last formula is the definition of the squared Mahalanobis distance. The derivation uses several matrix identities such as $(AB)^T = B^T A^T$, $(AB)^-1 = B^{-1}A^{-1}$, and $(A^{-1})^T = (A^T)^{-1}$. Notice that if Σ is the identity matrix, then the Mahalanobis distance reduces to the standard Euclidean distance between x and μ. The Mahalanobis distance accounts for the variance of each variable and the covariance between variables. Geometrically, it does this by transforming the data into standardized uncorrelated data and computing the ordinary Euclidean distance for the transformed data. In this way, the Mahalanobis distance is like a univariate z-score: it provides a way to measure distances that takes into account the scale of the data.

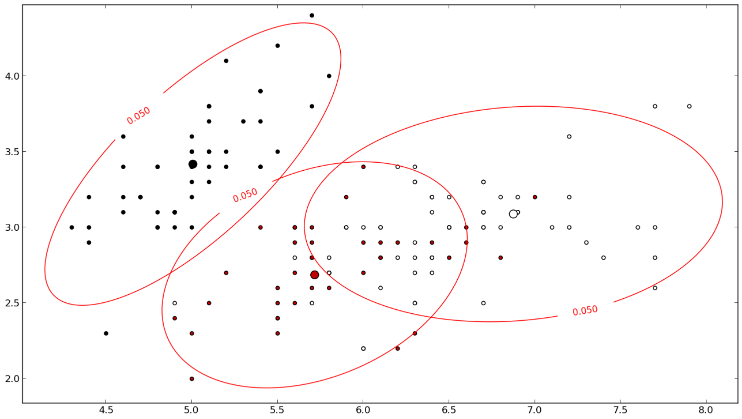

Figure 10: Mahalanobis distance is same for all ellipse in this figure

The figure indicates three different classes and the red line indicates the same Mahalanobis distance for each class. All points lying on the red line have the same distance from the class mean, because it is used the covariance matrix. The key feature is the use of covariance as a normalization factor.

7.Page Ranking Problem

The nodes 4 & 5 are dangling nodes in this case.The link structure gives matrix A as

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0.33 & 0 & 0 & 0 & 0 \\ 0.33 & 1 & 0 & 0 & 0 \\ 0.33 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 \end{pmatrix}$$

we form matrix *M=(1-p)*A+p*S* where S is a 5*5 matrix with all entries as $\frac{1}{n}$ where n is the number of pages in the web structure.

M matrix computed by taking damping factor p=0.15 is :

$$\begin{pmatrix} 0.030 & 0.030 & 0.030 & 0.200 & 0.200 \\ 0.313 & 0.030 & 0.030 & 0.200 & 0.200 \\ 0.313 & 0.880 & 0.030 & 0.200 & 0.200 \\ 0.313 & 0.030 & 0.455 & 0.200 & 0.200 \\ 0.030 & 0.030 & 0.455 & 0.200 & 0.200 \end{pmatrix}$$

The page rank obtained by computing eigen vector corresponding to eigen value 1 is:

$$\begin{pmatrix} 0.23857 \\ 0.30609 \\ 0.56630 \\ 0.54680 \\ 0.47928 \end{pmatrix}$$

Page rank obtained by power method with initial rank vector V=[0.2 0.2 0.2 0.2 0.2]^T is:

$$\begin{pmatrix} 0.11149 \\ 0.14305 \\ 0.26466 \\ 0.25553 \\ 0.22397 \end{pmatrix}$$

References

[1] NLP-stanford, *http://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html*

[2] Cluster Analysis wikipedia, *http://en.wikipedia.org/wiki/Cluster_analysis*.

[3] Advanced visualization techniques for Self-Organizing Maps with graph-based methods, *Georg P Olzlbauer, Andreas Rauber, and Michael Dittenbach*

[4] SOM Wikipedia, *http://en.wikipedia.org/wiki/Self-organizing_map*